NUMBER GUESSING GAME

Goal

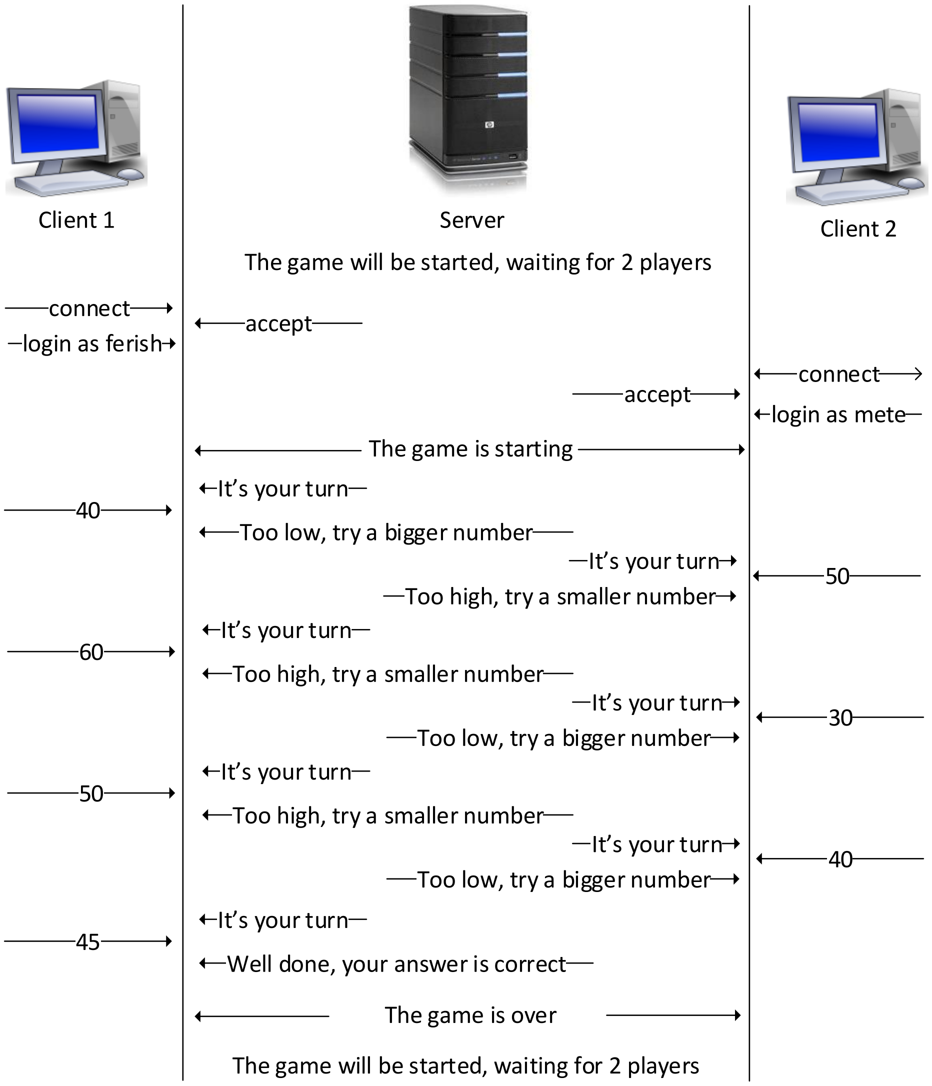
In this assignment, you are expected to design and implement a client-server application for a multi-player number guessing game by using POSIX sockets in C. The server application randomly selects a number from 1 to 99. Clients have to guess the number by making guesses until one of them finds the chosen number.

Scenario

You will implement a client - server application (client.c, server.c). Server and client programs will start only with the program's name. In one terminal, server will be started and in other terminals clients will be started. New instances of terminals will be used to start new clients.

You should define a variable (e.g. #define playerCount 2) to test the application with variable player counts. The minimum number of players (clients) is 2 whereas there must be only one server. All client terminals and server terminal will be running in the same Debian machine.

The server application is to be continuous (should not end unless it is stopped or killed). It waits for the defined number of client connection to start a new game. It manages the game and give a turn to each player (client) to guess a number. The game ends when one of the players finds the chosen number. Then, the server waits for new players to start a new game. Socket messaging between the server and the clients is illustrated in following figure for the game with two-players.


Client 1
Server
Client 2
The game will be started, waiting for 2 players
connect
accept
login as ferish
connect
accept
login as mete
The game is starting
It's your turn
40
Too low, try a bigger number
It's your turn
50
Too high, try a smaller number
It's your turn
60
Too high, try a smaller number
It's your turn
30
Too low, try a bigger number
It's your turn
50
Too high, try a smaller number
It's your turn
40
Too low, try a bigger number
It's your turn
45
Well done, your answer is correct
The game is over
The game will be started, waiting for 2 players

Server Interface

```
cme2002@cme:~/Desktop$ ./server

The game will be started, waiting for 2 players
Connection accepted
ferish logged in
Connection accepted
mete logged in

The game is starting

Round 1, ferish's turn
ferish guessed 40
Round 1, mete's turn
mete guessed 50

Round 2, ferish's turn
ferish guessed 60
Round 2, mete's turn
mete guessed 30

Round 3, ferish's turn
ferish guessed 50
Round 3, mete's turn
mete guessed 40

Round 4, ferish's turn
ferish guessed 45

ferish won the game

The game will be started, waiting for 2 players
```

Client Interface

Client 1:

```
cme2002@cme:~/Desktop$ ./client

Connected
Welcome to the number guessing game, please login to play
Player name: ferish

The game is starting

It’s your turn
Guess a number: 40
Too low, try a bigger number

It’s your turn
Guess a number: 60
Too high, try a smaller number

It’s your turn
Guess a number: 50
Too high, try a smaller number

It’s your turn
Guess a number: 45
Well done, your answer is correct

The game is over
```

Client 2:

```
cme2002@cme:~/Desktop/opsis/assignment$ ./client

Connected
Welcome to the number guessing game, please login to play
Player name: mete
The game is starting

It’s your turn
Guess a number: 50
Too high, try a smaller number

It’s your turn
Guess a number: 30
Too low, try a bigger number

It’s your turn
Guess a number: 40
Too low, try a bigger number

The game is over
```

Submission

- Submission will be via Google Classroom.
- Create two files: client.c, server.c
- Put client.c and server.c in a folder named exactly as your ***StudentID_Name_Surname*** (ex: 2017900000_Deniz_Yilmaz)
- Zip, or tar your folder and name it exactly as your StudentID_Name_Surname.tar (Ex: 2017900000_Deniz_Yilmaz.tar) You will lose **10 points** for not naming your submission properly.
- **-10 point** penalty apply to each day of late submissions.
- If your source files cannot be compiled in Debian you will get a 0. Compile your code before sending and make sure it works.
- If you submit the LAB codes that we give to you without any functionality required for this assignment you will get a 0. You cannot object as “I implemented socket connection, so I deserve some points”.
- Your submissions will be scanned among each other. Any assignments that are over the similarity threshold will get a 0 regardless of their similarity.

Due

3rd of December, 11:59 pm

For Questions

For any questions about the assignment, please send a mail to **feristah@cs.deu.edu.tr** with the subject “***OpSysAssignment2***” or comment under the assignment in Classroom.

Please read ALL of the instructions carefully!